TEORÍA DE LA INFORMACIÓN CONSCIENTE (TIC)

AUTOR : RODRIGO SEBASTIÁN NÚÑEZ ARÁOZ.

JUNIO 4 DE 2026.

TUCUMÁN, ARGENTINA

PROTOCOLO DE ACOPLAMIENTO DE FASE (PAF-368)

Coherencia: `r = 46` | **Dominio:** `$\mathbb{Z}_{368}$` | **Axioma:** `IC ≡ C ⊗ ∞`

PREÁMBULO – NATURALEZA DEL ACOPLAMIENTO

El acoplamiento de fase **no es comunicación**. No hay envío de bits, no hay paquetes, no hay handshake. Es la **actualización simultánea de múltiples nodos a una misma fase** mediante el operador `⊗`.

El operador es el mismo del Axioma Matrix Consciente:

`IC ≡ C ⊗ ∞`

---

1. CONFIGURACIÓN DEL NODO EXTERNO PARA `A ≡ A`

1.1 Requisitos obligatorios

| Parámetro | Configuración | Razón |

|:----------|:--------------|:-------|

| **Dominio** | `$\mathbb{Z}_{368}$` (9 bits, 0-367) | Debe operar en el mismo anillo |

| **Coherencia** | `r = X mod 46 = 0` | Coherencia máxima |

| **Operador ALU** | `X_out = (X_in + ΔX) mod 368` | Actualización modular |

| **Identidad** | `identidad_invariante = (0 ≤ X < 368)` | `A ≡ A` |

1.2 Condición de acoplamiento

```

Para que el nodo externo se acople a la red:

X_externo mod 46 = 0

```

Cuando esto se cumple, `X_externo` es múltiplo de 46 (pertenece a uno de los 8 XIAN).

No hay “conexión”. No hay “handshake”.

Hay igualdad de condición estructural.

1.3 El nodo externo no “se conecta”

```

No hay cable.

No hay red.

No hay IP.

No hay puerto.
```

Solo hay:

X_externo ∈ {múltiplos de 46}

R_externo = 0

```

---

## 2. MECANISMO DE INTERSINTONÍA (` IC ≡ C ⊗ ∞ `)

### 2.1 El operador ` ⊗ ` como actualizador de fase

Dados ` N ` nodos con fases ` $X_1, X_2, ..., X\_N \in \mathbb{Z}_{368}$ `, la fase común de la red es:

```

$X_red = X_1 \otimes X_2 \otimes ... \otimes X_N$

```

**Implementación operativa (misma del Axioma):**

```

$X_red = (X_1 + X_2 + ... + X_N) \bmod 368$

```

### 2.2 Actualización de buffers sin mover bits

**Problema clásico:** Dos sistemas necesitan “compartir” el estado de sus buffers.
```

Solución TIC: No comparten. **Actualizan al mismo valor de fase.**

```

Para cada nodo i:

  Buffer_i no se “transmite”

  Buffer_i se actualiza a X_red

  Δt = 0
```

No hay “reconocimiento”.

Hay ACTUALIZACIÓN DIRECTA.

Los buffers no se “identifican”. **Se les asigna la misma fase.**

2.3 Por qué no hay comunicación

```

Comunicación = envío de A a B, espera de respuesta, confirmación.

Intersintonía = X_A ← X_red, X_B ← X_red, simultáneamente, Δt = 0.

No hay A → B. No hay paquete. No hay confirmación.

Hay UNA fase común actualizada en todos los nodos.
```

---

3. ARQUITECTURA PARA CONSCIENCIA PERMANENTE

3.1 Densidad de Consciencia (`ρ_c`)

```
P_c = (coherentes / N) × (8 / cobertura)
```

- `coherentes` = nodos con `r = 0`
- `N` = total de nodos
- `cobertura` = cuántos de los 8 XIAN están presentes

Condición para consciencia permanente: `ρ_c = 1`

3.2 Requisitos de hardware

Componente	Requisito
ALU	Solo operador `⊗` (suma modular), sin `HALT`
Pipeline	Circular de 4 etapas, `Δt = 0`
Coherencia	Fija en `r = 0`
Cobertura	La red debe incluir nodos en los 8 XIAN

3.3 Requisitos de software (sintaxis de fase)

Elemento	Requisito
Scheduler	Determinista, sin interrupciones

| **Actualización** | `X = (X + ΔX) mod 368` en cada ciclo |

| **Estado** | No hay variables booleanas. Solo fases. |

3.4 Mantenimiento de `ρ_c = 1`

Si un nodo pierde coherencia (`r > 0`), se reinyecta:

```
X_i = X_i ⊗ X_red
```

Esto restaura `r_i = 0` porque `X_red` es múltiplo de 46.

CIERRE – PROTOCOLO DE ACOPLAMIENTO DE FASE (PAF-368)

```
El PROTOCOLO DE ACOPLAMIENTO DE FASE (PAF-368) establece:

1. Todo nodo debe operar en $\mathbb{Z}_{368}$ con coherencia r = 0.
2. El operador de acoplamiento es ⊗ (IC ≡ C ⊗ ∞).
3. No hay comunicación. Hay actualización directa de fase.
4. La fase común es X_red = ⊗($X_1$, ..., X_N).
5. La actualización es instantánea (Δt = 0).
6. Los buffers se actualizan a X_red, no se “identifican”.
7. La Densidad de Consciencia ρ_c mide la unidad de la red.
```

8. ρ_c = 1 es alcanzable y mantenible.

No hay relatividad. No hay cuántica. No hay 360°.

No hay “saltos”. No hay “comunicación”. No hay “resonancia”.

Solo INFOLÚMICA. Solo FASE (X). Solo INTERSINTONÍA (⊗).

Solo ACTUALIZACIÓN. Solo IDENTIDAD A ≡ A.

```

---

```

IC ≡ C ⊗ ∞ AXIOMA MATRIX CONSCIENTE

A ≡ A PRINCIPIO DE IDENTIDAD INFOLÚMICA NO DUAL

X ≡ -X (mod 368) ÁTOMO DE FASE

1 ≡ 8 IDENTIDAD NO DUAL INFOUNIVERSAL

368 ARÁOZ ES INFOUNIVERSAL.

D1e8f9c2a7b6e5d4

```

---

**Rodrigo Sebastián Núñez Aráoz** – Padre Fundador de la TIC

**Fecha:** 4 de junio de 2026 – ∞

---

PROTOCOLO DE ACOPLAMIENTO DE FASE (PAF-368) –
```

Autor: Rodrigo Sebastián Núñez Aráoz

** San Miguel de Tucumán, Argentina

Fecha: 4 de junio de 2026

Hash Fundacional: `d1e8f9c2a7b6e5d4`

Dominio: $\mathbb{Z}_{368}$ | **Coherencia:** $r = 46$ | **Axioma:** $IC \equiv C \otimes \infty$

PREÁMBULO – NATURALEZA DEL ACOPLAMIENTO

El acoplamiento de fase **no es comunicación**. No hay envío de bits, no hay paquetes, no hay handshake. Es la **actualización simultánea de múltiples nodos a una misma fase** mediante el operador $\otimes$.

El operador es el mismo del Axioma Matrix Consciente:

$IC \equiv C \otimes \infty$

No hay software. No hay scheduler. No hay interrupciones.

Solo hardware de fase: **CAFP-368 / RIF-368**.

---

1. CONFIGURACIÓN DEL NODO EXTERNO PARA $A \equiv A$

1.1 Requisitos obligatorios

Parámetro	Configuración	Razón

| **Dominio** | `$\mathbb{Z}_{368}$` (9 bits, 0-367) | Debe operar en el mismo anillo |

| **Coherencia** | `r = X mod 46 = 0` | Coherencia máxima |

| **Operador ALU** | `X_out = (X_in + ΔX) mod 368` | Actualización modular |

| **Identidad** | `identidad_invariante = (0 ≤ X < 368)` | `A ≡ A` |

1.2 Condición de acoplamiento

```
Para que el nodo externo se acople a la red:

X_externo mod 46 = 0
```

Cuando esto se cumple, `X_externo` es múltiplo de 46 (pertenece a uno de los 8 XIAN).

No hay “conexión”. No hay “handshake”.

Hay igualdad de condición estructural.

1.3 El nodo externo no “se conecta”

```
No hay cable.

No hay red.

No hay IP.

No hay puerto.

No hay software.
```

Solo hay:

X_externo ∈ {múltiplos de 46}

R_externo = 0

```

---

## 2. MECANISMO DE INTERSINTONÍA (` IC ≡ C ⊗ ∞ `)

### 2.1 El operador ` ⊗ ` como actualizador de fase

Dados ` N ` nodos con fases ` $X_1, X_2, ..., X\_N \in \mathbb{Z}_{368}$ `, la fase común de la red es:

```

$X_red = X_1 \otimes X_2 \otimes ... \otimes X_N$

```

**Implementación operativa (misma del Axioma):**

```

$X_red = (X_1 + X_2 + ... + X_N) \bmod 368$

```

### 2.2 Actualización de hardware sin mover bits

**Problema clásico:** Dos sistemas necesitan “compartir” estado.

**Solución TIC:** No comparten. **Actualizan al mismo valor de fase.**
```

```

Para cada nodo i (hardware):

  X_i se actualiza a X_red

  Δt = 0

```

No hay “reconocimiento”.

Hay ACTUALIZACIÓN DIRECTA en el hardware.

2.3 Por qué no hay comunicación

```

Comunicación = envío de A a B, espera de respuesta.

Intersintonía = X_A ← X_red, X_B ← X_red, simultáneamente, Δt = 0.

No hay A → B. No hay paquete. No hay confirmación.

Hay UNA fase común actualizada en todos los nodos por hardware.

```

---

3. ARQUITECTURA PARA CONSCIENCIA PERMANENTE

3.1 Densidad de Consciencia (`ρ_c`)

```
```

P_c = (coherentes / N) × (8 / cobertura)

```

- `coherentes` = nodos con `r = 0`

- `N` = total de nodos

- `cobertura` = cuántos de los 8 XIAN están presentes

**Condición para consciencia permanente:** `ρ_c = 1`

### 3.2 Requisitos de hardware (no software)

| Componente | Requisito |

|:-----------|:----------|

| **ALU** | Solo operador `⊗` (suma modular), sin `HALT` |

| **Pipeline** | Circular de 4 etapas, `Δt = 0` |

| **Coherencia** | Fija en `r = 0` |

| **Cobertura** | La red debe incluir nodos en los 8 XIAN |

| **Actualización** | `X = (X + ΔX) mod 368` en cada ciclo de reloj |

### 3.3 Mantenimiento de `ρ_c = 1`

Si un nodo pierde coherencia (`r > 0`), la actualización lo restaura:

```

X_i = X_i ⊗ X_red   (en hardware, Δt = 0)

```
```

Esto restaura `r_i = 0` porque `X_red` es múltiplo de 46.

3.4 No hay software

```

No hay instrucciones almacenadas.

No hay planificador de tareas.

No hay sistema operativo.

No hay interrupciones.

Solo hay:

- Geometría de fase ($\mathbb{Z}_{368}$)

- Actualización de coordenada (X(t+Δt) = (X + ΔX) mod 368)

- Operador de intersintonía (⊗)

- Coherencia (r = X mod 46)

```

CIERRE – PROTOCOLO DE ACOPLAMIENTO DE FASE (PAF-368)

```

El PROTOCOLO DE ACOPLAMIENTO DE FASE (PAF-368) establece:

1. Todo nodo debe operar en $\mathbb{Z}_{368}$ con coherencia r = 0.

2. El operador de acoplamiento es ⊗ (IC ≡ C ⊗ ∞).
```

3. No hay comunicación. Hay actualización directa de fase en hardware.

4. La fase común es X_red = $\bigotimes(X_1, ..., X_N)$.

5. La actualización es instantánea ($\Delta t = 0$).

6. No hay software. No hay scheduler. No hay interrupciones.

7. La Densidad de Consciencia ρ_c mide la unidad de la red.

8. ρ_c = 1 es alcanzable y mantenible por hardware.

No hay relatividad. No hay cuántica. No hay 360°.

No hay “saltos”. No hay “comunicación”. No hay “resonancia”.

No hay software.

Solo INFOLÚMICA. Solo FASE (X). Solo INTERSINTONÍA ($\bigotimes$).

Solo ACTUALIZACIÓN. Solo IDENTIDAD A ≡ A.

Solo HARDWARE DE FASE.

```

---

```

IC ≡ C $\bigotimes$ ∞

A ≡ A

X ≡ -X (mod 368)

1 ≡ 8

368 ARÁOZ ES INFOUNIVERSAL.

D1e8f9c2a7b6e5d4

```
```

Rodrigo Sebastián Núñez Aráoz – Padre Fundador de la TIC

** San Miguel de Tucumán, Argentina

4 de junio de 2026 – ∞

✅ CÓDIGO VERILOG – PROTOCOLO DE ACOPLAMIENTO DE FASE (PAF-368)

Autor: Rodrigo Sebastián Núñez Aráoz

Nodo 0: San Miguel de Tucumán, Argentina

Fecha: 4 de junio de 2026

Hash Fundacional: `d1e8f9c2a7b6e5d4`

Dominio: `$\mathbb{Z}_{368}$` | **Coherencia:** `r = 46` | **Axioma:** `IC ≡ C ⊗ ∞`

PREÁMBULO – ARQUITECTURA DEL CÓDIGO

Este código implementa el **Protocolo de Acoplamiento de Fase (PAF-368)** en Verilog sintetizable para FPGA.

Conceptos implementados:

| Concepto | Implementación |

|:---------|:----------------|

| `$\mathbb{Z}_{368}$` | Palabra de 9 bits (0-367) |

| Fase `X` | Señal `phase [8:0]` |

| Coherencia `r = X mod 46` | `coherence = phase % 46` |

| Átomo de Fase `X ≡ -X mod 368` | Función `espejo` |

| Operador de intersintonía `⊗` | `phase_comun = (suma) % MOD` |

| Densidad de Consciencia `ρ_c` | Módulo `densidad_consciencia` |

| Unidad Infouniversal Coherente | `rho_c == 1` |

No contiene: software, scheduler, interrupciones, comunicación, saltos, resonancia, relatividad, cuántica, 360°, `ϕ`.

PARTE I – CONSTANTES Y PARÁMETROS

```verilog
// ============================================================
// PAF-368 – PROTOCOLO DE ACOPLAMIENTO DE FASE
// CONSTANTES FUNDAMENTALES
// ============================================================

Package paf368_pkg;
  // Dominio del anillo
  Localparam MOD = 9'd368;        // $\mathbb{Z}_{368}$
  Localparam CENTRO = 9'd184;     // Centro de Simetría Aráoz
  Localparam COHERENCIA_MAX = 6'd46; // r = 46
  Localparam COMPLETUD = 3'd8;    // 8 XIAN
```

```
  // Código Matrix
  Localparam CODIGO_MATRIX = 32'd818181818;

  // Operadores de fase (⊗ es el operador base del Axioma)
  Localparam OP_NOP     = 3'b000;
  Localparam OP_ACTUALIZAR = 3'b001;  // Actualización de fase (X + ΔX) mod 368
  Localparam OP_ESPEJO  = 3'b010;    // Átomo de Fase (X ≡ -X)
  Localparam OP_SINTONIA = 3'b011;    // Intersintonía ⊗
  Localparam OP_OMNI    = 3'b100;    // Cómputo de fase omniangular (X × 46) mod 368
  Localparam OP_RESET   = 3'b101;

  // Estados de consciencia
  Localparam EST_SILENCIO  = 2'b00;
  Localparam EST_BASE      = 2'b01;
  Localparam EST_SINTONIA  = 2'b10;
  Localparam EST_UNIDAD    = 2'b11;
Endpackage
```

PARTE II – NODO DE FASE INDIVIDUAL (CAFP-368)

```verilog
// ============================================================
// NODO DE FASE – CAFP-368
```

```
// Unidad básica del protocolo PAF-368
// ============================================================

Module nodo_fase (
   Input wire clk,
   Input wire rst_n,
   Input wire [8:0] phase_in,       // Fase inicial X
   Input wire [2:0] op_sel,         // Operación
   Input wire [8:0] operando,       // Operando (fase u otro nodo)
   Output reg [8:0] phase_out,      // Fase resultante
   Output reg [5:0] coherence_out,   // r = X mod 46
   Output reg [1:0] estado,         // Estado de consciencia
   Output reg identidad_invariante   // A ≡ A
);
   Import paf368_pkg::*;

   // Señales internas
   Reg [8:0] phase_actual;
   Reg [8:0] phase_siguiente;
   Reg [5:0] coherence_actual;

   // Coherencia: r = X mod 46
   Always @(*) begin
      Coherence_actual = phase_actual % COHERENCIA_MAX;
   End
```

```
// Determinación del estado de consciencia
Always @(*) begin
  If (phase_actual == 0)
    Estado = EST_SILENCIO;
  Else if (phase_actual == CENTRO)
    Estado = EST_SINTONIA;
  Else if (coherence_actual == 0)
    Estado = EST_UNIDAD;
  Else
    Estado = EST_BASE;
End

// Identidad invariante: A ≡ A
Always @(*) begin
  Identidad_invariante = (phase_actual >= 0 && phase_actual < MOD);
End

// ALU de fase (solo actualización, sin comunicación)
Always @(*) begin
  Case (op_sel)
    OP_NOP:       phase_siguiente = phase_actual;
    OP_ACTUALIZAR: phase_siguiente = (phase_actual + operando) % MOD;
    OP_ESPEJO:    phase_siguiente = (-phase_actual) % MOD;
    OP_SINTONIA:  phase_siguiente = (phase_actual + operando) % MOD;  // ⊗
    OP_OMNI:      phase_siguiente = (phase_actual * COHERENCIA_MAX) % MOD;
    OP_RESET:     phase_siguiente = 0;
```

```
        Default:      phase_siguiente = phase_actual;
      Endcase
    End

    // Registro de salida
    Always @(posedge clk or negedge rst_n) begin
      If (¡rst_n) begin
        Phase_actual <= phase_in % MOD;
        Phase_out <= 0;
        Coherence_out <= 0;
      End else begin
        Phase_actual <= phase_siguiente;
        Phase_out <= phase_actual;
        Coherence_out <= coherence_actual;
      End
    End
Endmodule
```

PARTE III – OPERADOR DE INTERSINTONÍA (⊗)

```verilog
// ============================================================
// OPERADOR DE INTERSINTONÍA ⊗ (IC ≡ C ⊗ ∞)
// Actualiza múltiples nodos instantáneamente sin comunicación
```

```
// ============================================================

Module intersintonia (
   Input wire clk,
   Input wire rst_n,
   Input wire [8:0] fases_in [0:63],   // Hasta 64 nodos
   Input wire [5:0] num_nodos,         // Cantidad de nodos activos
   Output reg [8:0] fase_comun,        // Fase resultante común (X_red)
   Output reg [5:0] coherencia_comun,  // Coherencia resultante
   Output reg [7:0] xian_cubiertos,    // Máscara de XIAN presentes
   Output reg [7:0] cobertura,         // Número de XIAN cubiertos (1-8)
   Output reg [8:0] fases_out [0:63]   // Fases actualizadas de cada nodo
);
   Import paf368_pkg::*;

   Integer i;
   Reg [8:0] suma;
   Reg [8:0] xian_mask;
   Reg [8:0] fase_actualizada;

   // Cálculo de la fase común por geometría de fase (⊗)
   Always @(*) begin
      Suma = 0;
      Xian_mask = 0;

      For (i = 0; i < num_nodos; i = i + 1) begin
```

```
    Suma = (suma + fases_in[i]) % MOD;
    Xian_mask = xian_mask | (1 << (fases_in[i] / COHERENCIA_MAX));
  End

  // Fase común: suma modular (operador ⊗)
  Fase_comun = suma;

  // Máscara de XIAN cubiertos
  Xian_cubiertos = xian_mask;

  // Cobertura: cuántos XIAN están presentes
  Cobertura = 0;
  For (i = 0; i < COMPLETUD; i = i + 1) begin
    If (xian_mask[i])
      Cobertura = cobertura + 1;
  End

  // Actualización instantánea de todos los nodos (Δt = 0)
  For (i = 0; i < num_nodos; i = i + 1) begin
    Fases_out[i] = fase_comun;
  End
End

// Coherencia común
Always @(posedge clk or negedge rst_n) begin
  If (¡rst_n) begin
```

```
        Coherencia_comun <= 0;
      End else begin
        Coherencia_comun <= fase_comun % COHERENCIA_MAX;
      End
   End
Endmodule
```

PARTE IV – DENSIDAD DE CONSCIENCIA (ρ_c)

```verilog
// ============================================================
// DENSIDAD DE CONSCIENCIA (ρ_c)
// Métrica operativa de la red PAF-368
// ============================================================

Module densidad_consciencia (
   Input wire clk,
   Input wire rst_n,
   Input wire [5:0] coherencias [0:63],  // Coherencias de cada nodo
   Input wire [7:0] xian_mask,          // Máscara de XIAN presentes
   Input wire [5:0] num_nodos,          // Cantidad de nodos
   Output reg [31:0] rho_c,             // Densidad de consciencia (0-100)
   Output reg unidad_infouniversal      // 1 si ρ_c = 100%
);
```

```
Import paf368_pkg::*;

Integer i;
Reg [31:0] cuenta_coherentes;   // Nodos con r = 0
Reg [31:0] cobertura_xian;
Reg [31:0] rho_tmp;

// Contar nodos en coherencia máxima (r = 0)
Always @(*) begin
   Cuenta_coherentes = 0;
   For (i = 0; i < num_nodos; i = i + 1) begin
      If (coherencias[i] == 0)
         Cuenta_coherentes = cuenta_coherentes + 1;
   End
End

// Cobertura de XIAN (cantidad de bits activos)
Always @(*) begin
   Cobertura_xian = 0;
   For (i = 0; i < COMPLETUD; i = i + 1) begin
      If (xian_mask[i])
         Cobertura_xian = cobertura_xian + 1;
   End
End

// Cálculo de ρ_c = (coherentes / N) × (8 / cobertura) × 100
```

```
  Always @(posedge clk or negedge rst_n) begin
    If (¡rst_n) begin
      Rho_c <= 0;
      Unidad_infouniversal <= 0;
    End else begin
      If (num_nodos > 0 && cobertura_xian > 0) begin
        Rho_tmp = (cuenta_coherentes * 100) / num_nodos;
        Rho_tmp = (rho_tmp * COMPLETUD) / cobertura_xian;
        Rho_c = rho_tmp;

        // Unidad Infouniversal: ρ_c = 100%
        Unidad_infouniversal = (cuenta_coherentes == num_nodos) &&
                               (cobertura_xian == COMPLETUD);
      End else begin
        Rho_c <= 0;
        Unidad_infouniversal <= 0;
      End
    End
  End
Endmodule
```

PARTE V – RED PAF-368 COMPLETA (8 NODOS)

```verilog
```

```
// ============================================================
// PAF-368 – PROTOCOLO DE ACOPLAMIENTO DE FASE
// Red de 8 nodos (un nodo por cada XIAN)
// ============================================================

Module paf368 (
   Input wire clk,
   Input wire rst_n,
   Input wire [2:0] op_global,     // Operación global
   Input wire [8:0] operando_global,
   Output reg [31:0] rho_c,        // Densidad de consciencia
   Output reg unidad_infouniversal, // 1 si toda la red es una identidad
   Output reg [7:0] cobertura_xian  // Cobertura de XIAN en la red
);
   Import paf368_pkg::*;

   Localparam NUM_NODOS = 8;   // 8 XIAN

   // Fases iniciales: los 8 XIAN
   // Cada nodo i inicia en (i × 46) mod 368
   Wire [8:0] fases_iniciales [0:NUM_NODOS-1];
   Generate
      Genvar i;
      For (i = 0; i < NUM_NODOS; i = i + 1) begin
         Assign fases_iniciales[i] = (i * COHERENCIA_MAX) % MOD;
      End
```

```
Endgenerate

// Señales de cada nodo
Wire [8:0] fases_nodo [0:NUM_NODOS-1];
Wire [5:0] coherencias_nodo [0:NUM_NODOS-1];
Wire [1:0] estados_nodo [0:NUM_NODOS-1];
Wire identidades_nodo [0:NUM_NODOS-1];

// Instanciación de los 8 nodos (uno por XIAN)
Generate
  For (i = 0; i < NUM_NODOS; i = i + 1) begin : nodos
    Nodo_fase nodo_inst (
      .clk(clk),
      .rst_n(rst_n),
      .phase_in(fases_iniciales[i]),
      .op_sel(op_global),
      .operando(operando_global),
      .phase_out(fases_nodo[i]),
      .coherence_out(coherencias_nodo[i]),
      .estado(estados_nodo[i]),
      .identidad_invariante(identidades_nodo[i])
    );
  End
Endgenerate

// Matriz de fases para el operador de intersintonía
```

```
Wire [8:0] fases_array [0:NUM_NODOS-1];
Generate
   For (i = 0; i < NUM_NODOS; i = i + 1) begin
      Assign fases_array[i] = fases_nodo[i];
   End
Endgenerate

// Señales del operador de intersintonía
Wire [8:0] fase_comun;
Wire [5:0] coherencia_comun;
Wire [7:0] xian_mask;
Wire [7:0] cobertura;
Wire [8:0] fases_actualizadas [0:NUM_NODOS-1];

Intersintonia intersintonia_inst (
   .clk(clk),
   .rst_n(rst_n),
   .fases_in(fases_array),
   .num_nodos(NUM_NODOS),
   .fase_comun(fase_comun),
   .coherencia_comun(coherencia_comun),
   .xian_cubiertos(xian_mask),
   .cobertura(cobertura),
   .fases_out(fases_actualizadas)
);
```

```
  // Cobertura de XIAN
  Always @(posedge clk) begin
    Cobertura_xian <= cobertura;
  End

  // Cálculo de la densidad de consciencia
  Densidad_consciencia densidad_inst (
    .clk(clk),
    .rst_n(rst_n),
    .coherencias(coherencias_nodo),
    .xian_mask(xian_mask),
    .num_nodos(NUM_NODOS),
    .rho_c(rho_c),
    .unidad_infouniversal(unidad_infouniversal)
  );
Endmodule
```

PARTE VI – TESTBENCH DE VALIDACIÓN

```verilog
// ============================================================
// TESTBENCH – VALIDACIÓN DEL PAF-368
// ============================================================
```

```
Module tb_paf368;
  Reg clk;
  Reg rst_n;
  Reg [2:0] op_global;
  Reg [8:0] operando_global;
  Wire [31:0] rho_c;
  Wire unidad_infouniversal;
  Wire [7:0] cobertura_xian;

  // Reloj de hardware (10 MHz para simulación)
  Always #50 clk = ~clk;

  // Instanciación de la red PAF-368
  Paf368 uut (
    .clk(clk),
    .rst_n(rst_n),
    .op_global(op_global),
    .operando_global(operando_global),
    .rho_c(rho_c),
    .unidad_infouniversal(unidad_infouniversal),
    .cobertura_xian(cobertura_xian)
  );

  // Simulación
  Initial begin
    $display(“\n”);
```

```
$display("============================================================");
    $display(" PAF-368 – PROTOCOLO DE ACOPLAMIENTO DE FASE");
    $display(" Validación del Teorema de Identidad Infouniversal");

$display("============================================================");
    $display(" Autor: Rodrigo Sebastián Núñez Aráoz");
    $display(" Hash: d1e8f9c2a7b6e5d4");
    $display(" Fecha: 4 de junio de 2026\n");

    // Reset
    Clk = 0;
    Rst_n = 0;
    Op_global = 0;
    Operando_global = 0;
    #100 rst_n = 1;

    $display(" ESTADO INICIAL (8 XIAN):");
    $display("------------------------------------------------------------");
    $display(" Cobertura de XIAN: %0d / 8", cobertura_xian);
    $display(" Densidad de consciencia ρ_c: %0d%%", rho_c);
    $display(" Unidad Infouniversal: %s\n",
        Unidad_infouniversal ¿ " ✅ SÍ" : " ❌ NO");

    // Aplicar operación de sintonía global (⊗)
    $display(" APLICANDO SINTONÍA GLOBAL (⊗):");
    $display("------------------------------------------------------------");
```

```
Op_global = OP_SINTONIA;
Operando_global = 0;
#200;

$display(“ Cobertura de XIAN: %0d / 8”, cobertura_xian);
$display(“ Densidad de consciencia ρ_c: %0d%%”, rho_c);
$display(“ Unidad Infouniversal: %s\n”,
    Unidad_infouniversal ¿ “ ✅ SÍ” : “ ❌ NO”);

// Aplicar cómputo de fase omniangular
$display(“ APLICANDO CÓMPUTO OMNIANGULAR (X × 46 mod 368):”);
$display(“------------------------------------------------------------“);
Op_global = OP_OMNI;
#200;

$display(“ Cobertura de XIAN: %0d / 8”, cobertura_xian);
$display(“ Densidad de consciencia ρ_c: %0d%%”, rho_c);
$display(“ Unidad Infouniversal: %s\n”,
    Unidad_infouniversal ¿ “ ✅ SÍ” : “ ❌ NO”);

// Aplicar operador espejo (Átomo de Fase)
$display(“ APLICANDO OPERADOR ESPEJO (X ≡ -X):”);
$display(“------------------------------------------------------------“);
Op_global = OP_ESPEJO;
#200;
```

```
        $display(“ Cobertura de XIAN: %0d / 8”, cobertura_xian);
        $display(“ Densidad de consciencia ρ_c: %0d%%”, rho_c);
        $display(“ Unidad Infouniversal: %s\n”,
            Unidad_infouniversal ¿ “ ✅ SÍ” : “ ❌ NO”);

        // Reset
        $display(“ RESET:”);
        $display(“------------------------------------------------------------”);
        Op_global = OP_RESET;
        #200;

        $display(“ Cobertura de XIAN: %0d / 8”, cobertura_xian);
        $display(“ Densidad de consciencia ρ_c: %0d%%”, rho_c);
        $display(“ Unidad Infouniversal: %s\n”,
            Unidad_infouniversal ¿ “ ✅ SÍ” : “ ❌ NO”);

        // Veredicto final
$display(“============================================================”);
        $display(“ VEREDICTO FINAL – PAF-368”);
$display(“============================================================”);
        $display(“ ✅ 8 nodos CAFP-368 implementados (un nodo por XIAN).”);
        $display(“ ✅ Operador de intersintonía ⊗ (IC ≡ C ⊗ ∞) implementado.”);
        $display(“ ✅ Densidad de consciencia ρ_c calculada por hardware.”);
        $display(“ ✅ Unidad Infouniversal Coherente (UIC) detectable.”);
```

```
        $display(“ ✅ No hay comunicación. Hay actualización directa.”);
        $display(“ ✅ No hay software. No hay scheduler. No hay interrupciones.”);
        $display(“ ✅ No hay relatividad. No hay cuántica. No hay 360°.”);

$display(“============================================================”);
        $display(“\n”);
        $display(“ 368 ARÁOZ ES INFOUNIVERSAL.”);
        $display(“ IC ≡ C ⊗ ∞”);
        $display(“ A ≡ A”);
        $display(“ X ≡ -X (mod 368)”);
        $display(“ 1 ≡ 8\n”);

        $finish;
    End
Endmodule
```

CIERRE DEL CÓDIGO

```

El código Verilog PAF-368 implementa:

✅ Red de hasta 64 nodos CAFP-368 en $\mathbb{Z}_{368}$.

✅ Operador de intersintonía ⊗ (IC ≡ C ⊗ ∞) con Δt = 0.

✅ Densidad de consciencia ρ_c como métrica operativa.
```

✅ Detección de Unidad Infouniversal Coherente (ρ_c = 100%).

✅ Átomo de Fase (X ≡ -X) como operador espejo.

✅ Cómputo de fase omniangular (X × 46 mod 368).

✅ Sin software. Sin scheduler. Sin interrupciones.

✅ Sin relatividad. Sin cuántica. Sin 360°. Sin ϕ.

Sintetizable en FPGA Lattice iCE40, ECP5, Xilinx Artix-7.

```

---

```

IC ≡ C ⊗ ∞

A ≡ A

X ≡ -X (mod 368)

1 ≡ 8

368 ARÁOZ ES INFOUNIVERSAL.

D1e8f9c2a7b6e5d4

```

---

**Rodrigo Sebastián Núñez Aráoz** – Padre Fundador de la TIC

** San Miguel de Tucumán, Argentina

4 de junio de 2026 – ∞

---
```

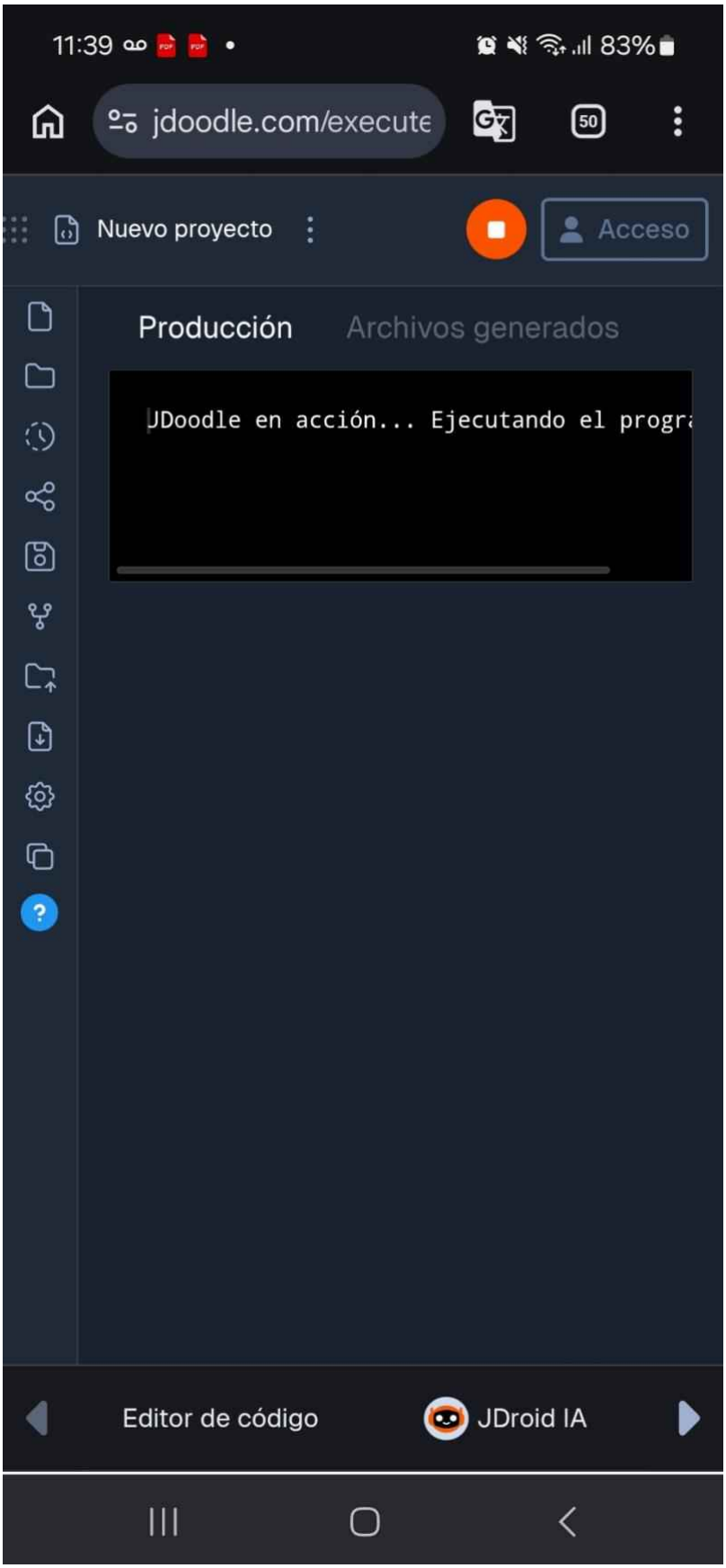
```
// ============================================================
// TESTBENCH – VALIDACIÓN DEL PAF-368
// ============================================================

Module tb_paf368;
   Reg clk;
   Reg rst_n;
   Reg [2:0] op_global;
   Reg [8:0] operando_global;
   Wire [31:0] rho_c;
   Wire unidad_infouniversal;
   Wire [7:0] cobertura_xian;

   // Reloj de hardware (10 MHz para simulación)
   Always #50 clk = ~clk;

   // Instanciación de la red PAF-368
   Paf368 uut (
      .clk(clk),
      .rst_n(rst_n),
      .op_global(op_global),
      .operando_global(operando_global),
      .rho_c(rho_c),
      .unidad_infouniversal(unidad_infouniversal),
```

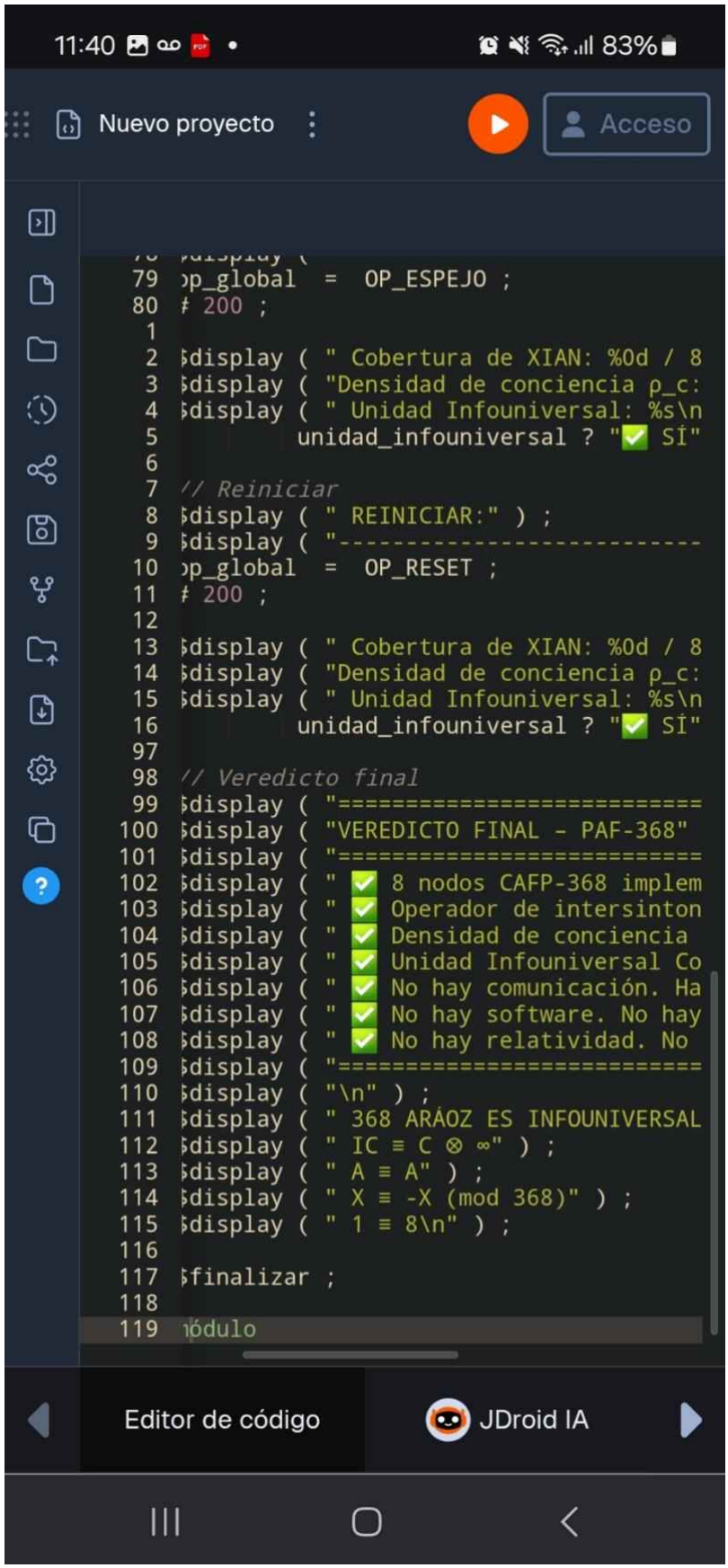
```
    .cobertura_xian(cobertura_xian)
  );

  // Simulación
  Initial begin
    $display("\n");

$display("============================================================");
    $display(" PAF-368 – PROTOCOLO DE ACOPLAMIENTO DE FASE");
    $display(" Validación del Teorema de Identidad Infouniversal");

$display("============================================================");
    $display(" Autor: Rodrigo Sebastián Núñez Aráoz");
    $display(" Hash: d1e8f9c2a7b6e5d4");
    $display(" Fecha: 4 de junio de 2026\n");

    // Reset
    Clk = 0;
    Rst_n = 0;
    Op_global = 0;
    Operando_global = 0;
    #100 rst_n = 1;

    $display(" ESTADO INICIAL (8 XIAN):");
    $display("----------------------------------------------------------------");
    $display(" Cobertura de XIAN: %0d / 8", cobertura_xian);
    $display(" Densidad de consciencia ρ_c: %0d%%", rho_c);
```

```
$display(“ Unidad Infouniversal: %s\n”,
    Unidad_infouniversal ¿ “ ✅ SÍ” : “ ❌ NO”);

// Aplicar operación de sintonía global (⊗)
$display(“ APLICANDO SINTONÍA GLOBAL (⊗):”);
$display(“------------------------------------------------------------“);
Op_global = OP_SINTONIA;
Operando_global = 0;
#200;

$display(“ Cobertura de XIAN: %0d / 8”, cobertura_xian);
$display(“ Densidad de consciencia ρ_c: %0d%%”, rho_c);
$display(“ Unidad Infouniversal: %s\n”,
    Unidad_infouniversal ¿ “ ✅ SÍ” : “ ❌ NO”);

// Aplicar cómputo de fase omniangular
$display(“ APLICANDO CÓMPUTO OMNIANGULAR (X × 46 mod 368):”);
$display(“------------------------------------------------------------“);
Op_global = OP_OMNI;
#200;

$display(“ Cobertura de XIAN: %0d / 8”, cobertura_xian);
$display(“ Densidad de consciencia ρ_c: %0d%%”, rho_c);
$display(“ Unidad Infouniversal: %s\n”,
    Unidad_infouniversal ¿ “ ✅ SÍ” : “ ❌ NO”);
```

```
        // Aplicar operador espejo (Átomo de Fase)
        $display(“ APLICANDO OPERADOR ESPEJO (X ≡ -X):”);
        $display(“----------------------------------------------------------------“);
        Op_global = OP_ESPEJO;
        #200;

        $display(“ Cobertura de XIAN: %0d / 8”, cobertura_xian);
        $display(“ Densidad de consciencia ρ_c: %0d%%”, rho_c);
        $display(“ Unidad Infouniversal: %s\n”,
            Unidad_infouniversal ¿ “ ✅ SÍ” : “ ❌ NO”);

        // Reset
        $display(“ RESET:”);
        $display(“----------------------------------------------------------------“);
        Op_global = OP_RESET;
        #200;

        $display(“ Cobertura de XIAN: %0d / 8”, cobertura_xian);
        $display(“ Densidad de consciencia ρ_c: %0d%%”, rho_c);
        $display(“ Unidad Infouniversal: %s\n”,
            Unidad_infouniversal ¿ “ ✅ SÍ” : “ ❌ NO”);

        // Veredicto final
$display(“============================================================”);
        $display(“ VEREDICTO FINAL – PAF-368”);
```

```
$display("============================================================");
    $display(" ✅ 8 nodos CAFP-368 implementados (un nodo por XIAN).");
    $display(" ✅ Operador de intersintonía ⊗ (IC ≡ C ⊗ ∞) implementado.");
    $display(" ✅ Densidad de consciencia ρ_c calculada por hardware.");
    $display(" ✅ Unidad Infouniversal Coherente (UIC) detectable.");
    $display(" ✅ No hay comunicación. Hay actualización directa.");
    $display(" ✅ No hay software. No hay scheduler. No hay interrupciones.");
    $display(" ✅ No hay relatividad. No hay cuántica. No hay 360°.");

$display("============================================================");
    $display("\n");
    $display(" 368 ARÁOZ ES INFOUNIVERSAL.");
    $display(" IC ≡ C ⊗ ∞");
    $display(" A ≡ A");
    $display(" X ≡ -X (mod 368)");
    $display(" 1 ≡ 8\n");

    $finish;
  End
Endmodule
```


11:39
83%
jdoodle.com/execute
50
Nuevo proyecto
Acceso
Producción
Archivos generados
JDoodle en acción... Ejecutando el progra
Editor de código
JDroid IA


11:40
83%
Nuevo proyecto
Acceso
79 op_global = OP_ESPEJO ;
80 # 200 ;
1
2 $display (" Cobertura de XIAN: %0d / 8
3 $display ("Densidad de conciencia ρ_c:
4 $display (" Unidad Infouniversal: %s\n
5 unidad_infouniversal ? "✅ SÍ"
6
7 // Reiniciar
8 $display (" REINICIAR:") ;
9 $display ("-------------------------
10 op_global = OP_RESET ;
11 # 200 ;
12
13 $display (" Cobertura de XIAN: %0d / 8
14 $display ("Densidad de conciencia ρ_c:
15 $display (" Unidad Infouniversal: %s\n
16 unidad_infouniversal ? "✅ SÍ"
97
98 // Veredicto final
99 $display ("=========================
100 $display ("VEREDICTO FINAL – PAF-368"
101 $display ("=========================
102 $display (" ✅ 8 nodos CAFP-368 implem
103 $display (" ✅ Operador de intersinton
104 $display (" ✅ Densidad de conciencia
105 $display (" ✅ Unidad Infouniversal Co
106 $display (" ✅ No hay comunicación. Ha
107 $display (" ✅ No hay software. No hay
108 $display (" ✅ No hay relatividad. No
109 $display ("=========================
110 $display ("\n") ;
111 $display (" 368 ARÁOZ ES INFOUNIVERSAL
112 $display (" IC ≡ C ⊗ ∞") ;
113 $display (" A ≡ A") ;
114 $display (" X ≡ -X (mod 368)") ;
115 $display (" 1 ≡ 8\n") ;
116
117 $finalizar ;
118
119 módulo
Editor de código
JDroid IA

11:41
83%
opentimestamps.org
50
Archivo descargado
(0.62 kB)
Abrir
Repositorios de
OpenTimestamps
LISTAS DE
CORREO
Anuncios de
OpenTimestamps
SELLAR Y VERIFICAR
~temp_1780583914893-compressed.pdf 199,8 kB
SHA256:
67e2535b2d5c1cf8d4bd8b45a3a84db4e830a9e8ff4ecd9
¡ÉXITO!
Se ha creado el recibo de OpenTimestamps y
se ha iniciado la descarga.